

# **FITUR *DISCOVERY* APLIKASI B.O.S.S (*BUILD OUR SPACE STORE*)**

**TUGAS BESAR MATA KULIAH  
METODE FORMAL  
SE-46-03**



Disusun Oleh:  
**KELOMPOK 11**

- |                           |              |
|---------------------------|--------------|
| 1. Naufal Ajhar El Hafizh | (1302223040) |
| 2. Haikal Alfaro          | (1302220098) |
| 3. M. Galang Pradwita     | (1302223015) |
| 4. Muhammad Rafi Nadhif   | (1302220142) |

**PROGRAM STUDI S1 REKAYASA PERANGKAT LUNAK  
FAKULTAS INFORMATIKA  
TELKOM UNIVERSITY  
2023/2024**

# BAB 1

## PENDAHULUAN

### A. Latar Belakang

Fitur discovery atau pencarian adalah fitur yang sangat dibutuhkan dalam sebuah *e-commerce* dan transaksi *online* guna memberikan pelayanan yang maksimal kepada pengguna sehingga pengguna merasa lebih dimudahkan dalam transaksi. Bila pengguna merasa nyaman dan mudah menggunakan aplikasi belanja onlinenya, maka dapat meningkatkan kepercayaan pengguna. Dengan demikian, dapat mendatangkan lebih banyak pengguna dan pada akhirnya akan meningkatkan pertumbuhan bisnis.

### B. Tujuan

Fitur discovery adalah fitur yang memungkinkan pengguna aplikasi B.O.S.S untuk menemukan produk-produk yang sesuai dengan kebutuhan dan minat mereka. Fitur ini dapat memberikan rekomendasi produk berdasarkan kriteria tertentu, seperti kategori, harga, lokasi, ulasan, dan lainnya. Fitur ini juga dapat menampilkan produk-produk terbaru, terlaris, atau terpopuler di aplikasi B.O.S.S.

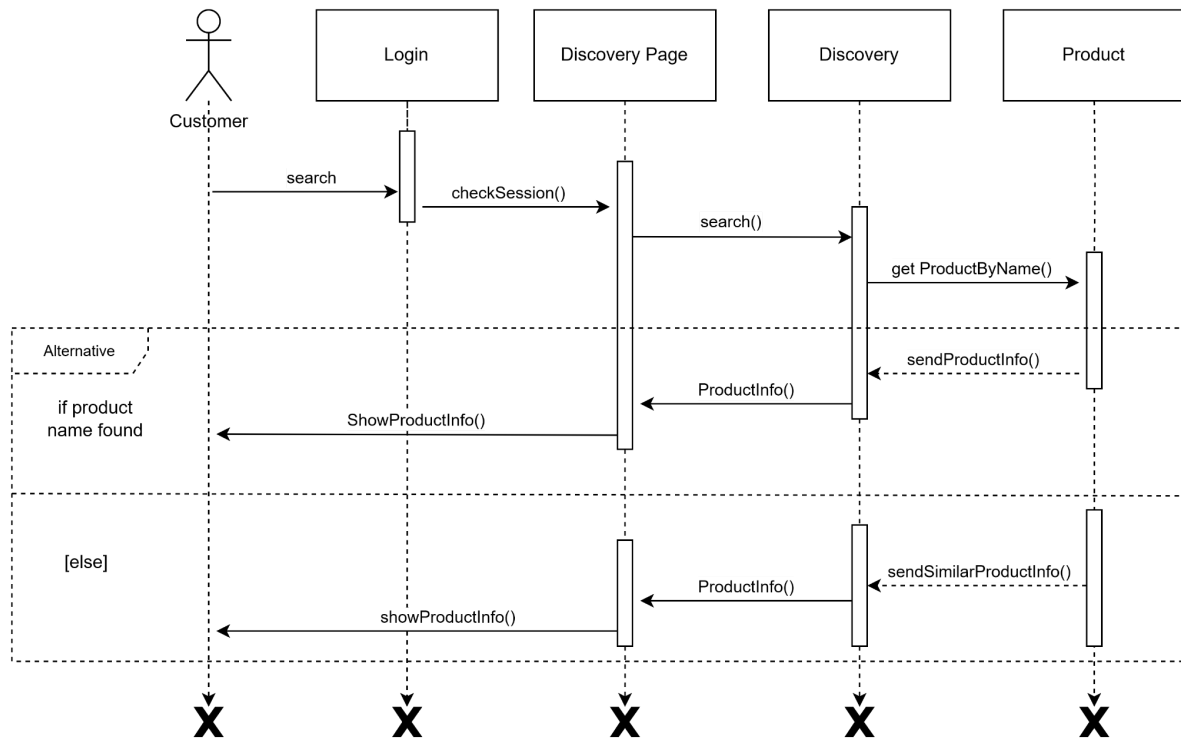
Tujuan dari fitur discovery adalah :

1. Meningkatkan pengalaman pengguna aplikasi toko bangunan, dengan memberikan kemudahan dan kecepatan dalam mencari dan memilih produk yang diinginkan.
2. Meningkatkan penjualan dan loyalitas pelanggan, dengan menawarkan produk-produk yang relevan, berkualitas, dan kompetitif.
3. Membantu pengguna aplikasi toko bangunan untuk menemukan solusi terbaik untuk kebutuhan bangunan mereka.
4. Meningkatkan kepuasan pengguna dengan memberikan rekomendasi produk yang sesuai dengan kebutuhan dan minat mereka.
5. Meningkatkan retensi pelanggan dengan menawarkan produk-produk yang terbaru, terlaris, atau terpopuler di aplikasi toko bangunan.

## C. Pemodelan

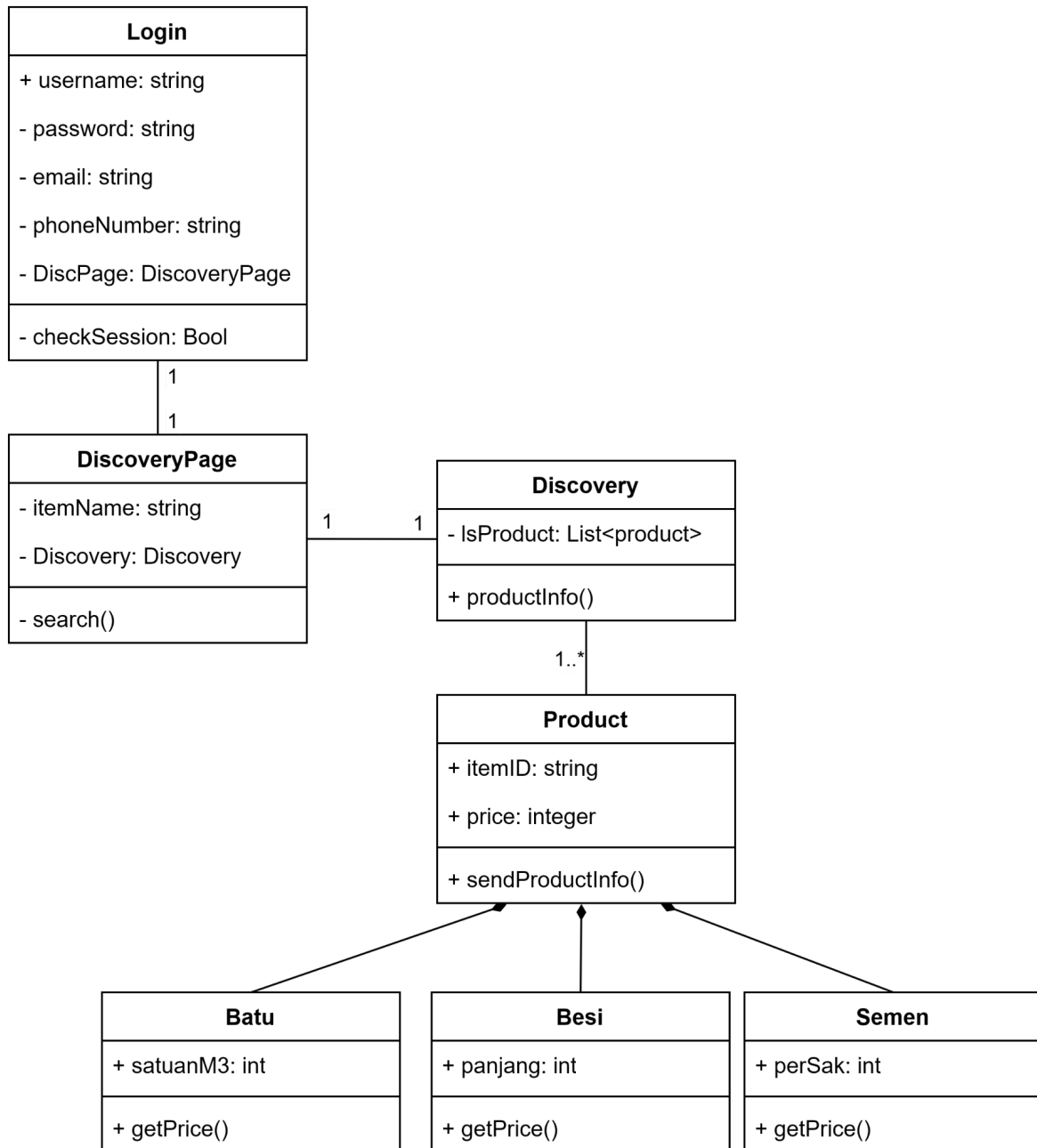
Kami menggunakan pemodelan UML berupa sequence diagram dan class diagram untuk sistem discovery aplikasi e-commerce :

### Sequence Diagram



Pengguna menekan discovery, kemudian sistem aplikasi akan mengarahkan ke discovery page sehingga pengguna dapat melakukan eksplorasi terkait barang yang mereka cari atau butuhkan. Pengguna dapat melakukan pencarian dengan menekan tombol search lalu sistem aplikasi akan memproses pencarian, jika barang yang dicari ditemukan maka akan langsung ditampilkan ke pengguna, jika tidak sistem aplikasi akan menampilkan saran barang yang berkaitan dengan pencarian si pengguna.

## Class Diagram



# Alloy

```
open util/ordering[Product]

sig login {
  username: String,
  password: String,
  email: String,
  phoneNumber: String,
  discPage: DiscoveryPage
}

sig DiscoveryPage {
  itemName: String,
  discovery: Discovery
}

sig Discovery {
  isProduct: Product -> lone Product
}

sig Product {
  itemId: String,
  price: Int
}

sig Batu {
  satuanM3: Int
} {
  all p: Product | p in Discovery.isProduct[this] implies p in this
}

sig Besi {
  panjang: Int
} {
  all p: Product | p in Discovery.isProduct[this] implies p in this
}

sig Semen {
  perSak: Int
} {
  all p: Product | p in Discovery.isProduct[this] implies p in this
}

-- Associations
fact loginAssociations {
  all l: login, dp: DiscoveryPage | l.discPage = dp
}

fact DiscoveryPageAssociations {
  all dp: DiscoveryPage, d: Discovery | dp.discovery = d
}

fact DiscoveryAssociations {
  all d: Discovery, p: Product | p in d.isProduct
}

-- Operations/Methods
pred checkSession[]: login[] {
  // Implementation of checkSession method
  !sessionStatus = "active"
}

pred search[dp: DiscoveryPage, itemNameToSearch: String] {
  // Implementation of search method
  some p: Product | p in dp.discovery.isProduct and p.itemId = itemNameToSearch
}

pred productInfo[p: Product] {
  // Implementation of productInfo method
  p in this.isProduct
}

pred getPrice[B: Batu, p: Product] {
  // Implementation of getPrice method for Batu
  p in this.isProduct
  harga batu per meter kubik * volume product
  let volume = p.price / b.satuanM3
  volume > 0
}

pred getPrice[B: Besi, p: Product] {
  // Implementation of getPrice method for Besi
  p in this.isProduct
  harga besi per meter * panjang product
  let totalPrice = b.panjang * p.price
  totalPrice > 0
}

pred getPrice[S: Semen, p: Product] {
  // Implementation of getPrice method for Semen
  p in this.isProduct
  harga semen per sak * jumlah sak product
  let totalPrice = s.perSak * p.price
  totalPrice > 0
}

run {} for 5 but 4 Product, 3 Batu, 3 Besi, 3 Semen
```